Mycat配置说明

环境参数

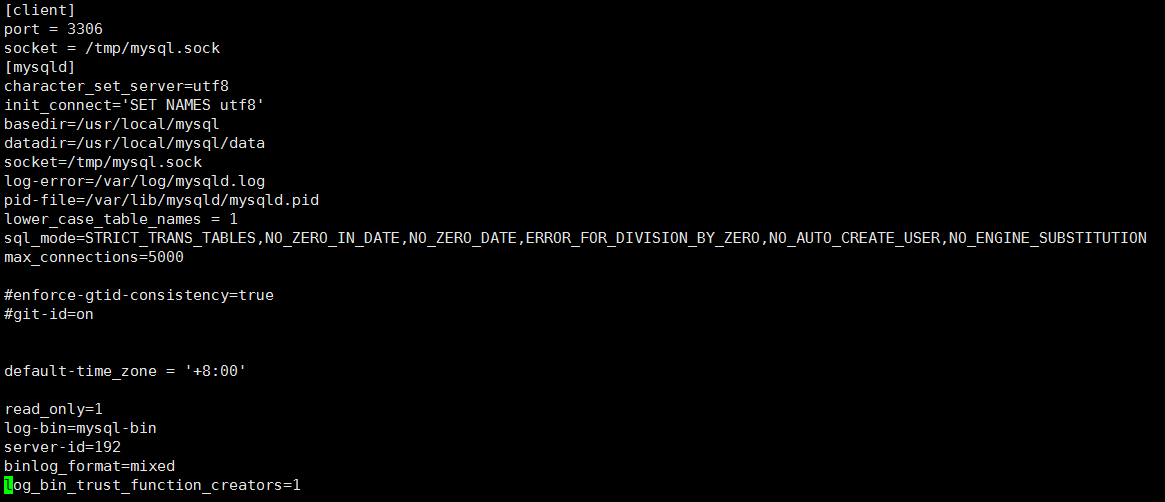
CentOS JDK1.7

MyCAT 1.6.5 MySql 5.7

MySQL 5.7.21安装

访问 <http://www.jb51.net/article/133835.htm> 安装说明

my.cnf



**5.7后修改密码:**

**update user set authentication\_string=password('123456') where user='root';**

flush privileges; #立即生效

**ps –ef**

**netstat -ntlp**

**ps aux | grep mycat**

MyCat的主要作用

实现后端数据库的读写分离及负载均衡

对业务数据库进行垂直切分

对业务数据库进行水平切分

控制数据库的连接数量

MyCAT的基本元素

逻辑库

相当于mysql中的数据库。

逻辑库可对应后端多个物理数据库。

逻辑库中并不保存数据。

逻辑表

相当于Mysql中的数据表

逻辑表可对应后端多个物理数据库中的表

逻辑表中并不保存数据

逻辑表的类别

分片表与非分片表按是否被分片划分

全局表，在所有分片之中都存在的表

ER关系表，按ER关系进行分片的表

Mycat的安装步骤：

1. 下载mycat

wget <http://dl.mycat.io/1.6.5/Mycat-server-1.6.5-release-20180122220033-linux.tar.gz>

tar zxf 解压

1. 安装jdk1.7 以上

rpm -qa | grep java 查看安装java

rpm -e --nodeps java 卸载查看java

1. 新建mycat运行系统账号

adduser mycat

chown mycat:mycat -R mycat/

1. 配置系统环境变量

查看服务器内存 free –m

vim /etc/profile

export PATH=$PATH:/usr/local/mycat/bin:/usr/local/java/jdk1.8

export JAVA\_HOME=/usr/local/java/jdk1.8

export CLASSPATH=.:$JAVA\_HOME/jre/lib/rt.jar:$JAVA\_HOME/lib.tools.jar

export MYCAT\_HOME=/usr/local/mycat

source /etc/profile

1. 修改mycat启动参数

mycat start

ps –ef

MyCAT配置文件

server.xml文件

配置系统相关参数

配置用户访问权限

配置sql防火墙及sql拦截功能

表级 DML 权限设置 代表 1 代表 insert update select delete

用户密码加密

进入到lib 下 执行

java -cp Mycat-server-1.6-RELEASE.jar io.mycat.util.DecryptUtil 0:root:123456

GO0bnFVWrAuFgr1JMuMZkvfDNyTpoiGU7n/Wlsa151CirHQnANVk3NzE3FErx8v6pAcO0ctX3xFecmSr+976QA==

需要加入<property name="usingDecrypt">1</property>

log4j2.xml 文件日志文件

rule.xml 文件

配置水平分片规则

配置分片规则所对应的分片函数

schema.xml 文件

table 定义数据库逻辑表

dataNode 定义数据存储的物理位置

database 属性定义物理数据库名称

dataHost 定义后端数据库主机信息

dataHosst balance 有4中取值：

0:不开启读写分离机制

1:全部的readHost与stand by writeHost 参与select语句的负载均衡

2:所有的readHost与writeHost都参与select语句的负载均衡

3: 所有的readHost 参与select语句的负载均衡

Mycat核心配置

Mycat 垂直分库

垂直分库的步骤

1. 收集分析业务模块间的关系
2. 复制数据库到其它实例

创建数据库命令 mysql –uroot –p –e “create database db”

1. 备份原数据并记录相关事务点

使用master-data=2 记录事务日志点

使用change master to 配置复制链路

使用change replication filter 配置数据库名转换

mysqldump --master-data=2 --single-transaction --routines --triggers --events -uroot -p smw > smwbak.sql

拷贝文件到其它服务器上

scp smwbak.sql [root@10.10.2.45:/root](mailto:root@10.10.2.45:/root)

导入sql文件

I need to database;

mysql -uroot -p smw < smwbak.sql

begin 主从复制：

在master 中创建主从复制账号

create user 'santint'@'10.10.3.%' identified by '123123';

授权

grant replication slave on \*.\* to 'santint'@'10.10.3.%';

flush privileges;

从库中创建链路

change master to master\_host=“10.10.2.45”,

master\_user=“zhangfan”,

master\_passwrod=”123456”,

master\_log\_file=”mysql-bin.000060”,

master\_log\_pos=123;

从库中查看链路 show slave status\G;

reset slave;

reset slave all;

mycat server:

MySql: reload @@config 更新配置文件，不需要重启mycat

创建过滤链接 \h change replication filter 查看命令

主从数据库可以不一致

change replication filter replicate\_rewrite\_db=((test\_db,order\_db));

1. 在原数据库中简历复制用户
2. 在新实例上恢复备份数据库
3. 在新实例上配置复制链路
4. 在新实例上启动复制
5. 配置MyCat垂直分库
6. 创建mycat用户

create user zhangfan @'10.10.2.%' identified by '123456';

授权

grant select,insert,update,delete on \*.\* to zhangfan'10.10.2.%';

1. 通过mycat访问DB

结束主从同步

stop slave;

reset slave all;

1. 删除原库中已迁移的表。

mysql –uroot –p

show variables like ‘read\_only';

set global read\_only=off;

垂直切分的优点

数据库的拆分简单明了，拆分规则明确。

应用程序模块清晰明确，整合容易

数据维护方便易行，容易定位。

缺点

部分表关联无法再数据库级别完成，需要在程序中完成。

访问频繁数据量大的表存在性能瓶颈

切分达到一定程度后，扩展性会遇到限制

解决跨分片关联方式

使用mycat全局表

冗余部分关键数据

使用api方式获取数据

MyCat水平分库

分片原则

能不切分尽量不要切分

选择合适的切分规则和分片键

尽量避免跨分片join操作

分片后如何处理查询 切分表

id % 3 = 0 -> db1

id % 3 = 1 -> db2

id % 3 = 2 -> db3

水平分片的步骤

1. 根据业务状态确定要进行水平切分的表。
2. 分析业务模型选择分片键及分片算法。
3. 使用mycat 部署分片集群。
4. 测试分片集群。
5. 业务及数据迁移。

如何选择分片键

1. 尽可能的比较均匀分布数据到各个节点上。
2. 该业务字段是最频繁的或者最重要的查询条件

分析业务模型选择分片键及分片算法

1. 对订单相关表进行水平切分
2. 以customer\_id做为分片键。
3. 采用简单取模分片算法。

使用MyCat部署分片集群

使用schema.xml 配置逻辑库及逻辑表

使用rule.xml 配置分片表的分片规则

使用server.xml 配置用户名和访问权限。

演示环境说明

|  |  |  |  |
| --- | --- | --- | --- |
| 主机名 | IP | 角色 | 数据库 |
| Node1 | 10.10.2.45 | MyCat |  |
| Node2 | 10.10.2.40 | MySql | Orderdb01  Orderdb02 |
| Node3 | 10.10.2.43 | Mysql | Orderdb03  Orderdb04 |

MyCat全局自增ID

1、创建mycat DB;

在mycat ./conf /dbseq.sql

2、导入 dbseq.sql mysql –uroot –p mysql < dbseq.sql

数据库为 MYCAT\_SEQUENCE;

3、使用全局自增id 需要改动server.xml文件，加入自增类型 0 1 2 3 4 五中类型

<property name=”sequnceHandlerType”>1</property>

0 代表本地文件方式 1 数据库方式 2时间戳方式 默认数据库方式1

增加数据节点

<dataNode name=”mycat” dataHost=”M1” database=”mycat”/>

4、修改sequence\_db\_conf.properties

ORDER\_MASTER=mycat

5、在MYCAT\_SEQUENCE 中加入记录 需要使用大写

insert into MYCAT\_SEQUENCE values(‘ORDER\_MASTER’,1,1);

1. 修改schema.xml 逻辑表启动自增id

<table autoIncrement=”true”></table> 开启自增ID

show grants for [zf@’10.10.2.%](mailto:zf@'10.10.2.%25)’; 查看权限

grant execute on \*.\* to ‘zf’@’10.10.3.%’; 授权限

python init\_order\_master.py 执行订单生成脚本

python init\_order.py 执行订单和详情脚本

ER分片表

关联时候 需要配置 字表父表

<childTable> 代表字表 加入到table 下

joinKey 代表关联key

<table>

<childTable name=”” primaryKey=”” joinKey=””

parentKey=”” autoIncrement=”true”/>

</table>

MyCAT 的其它常用功能

SQL拦截 配置在server.xml

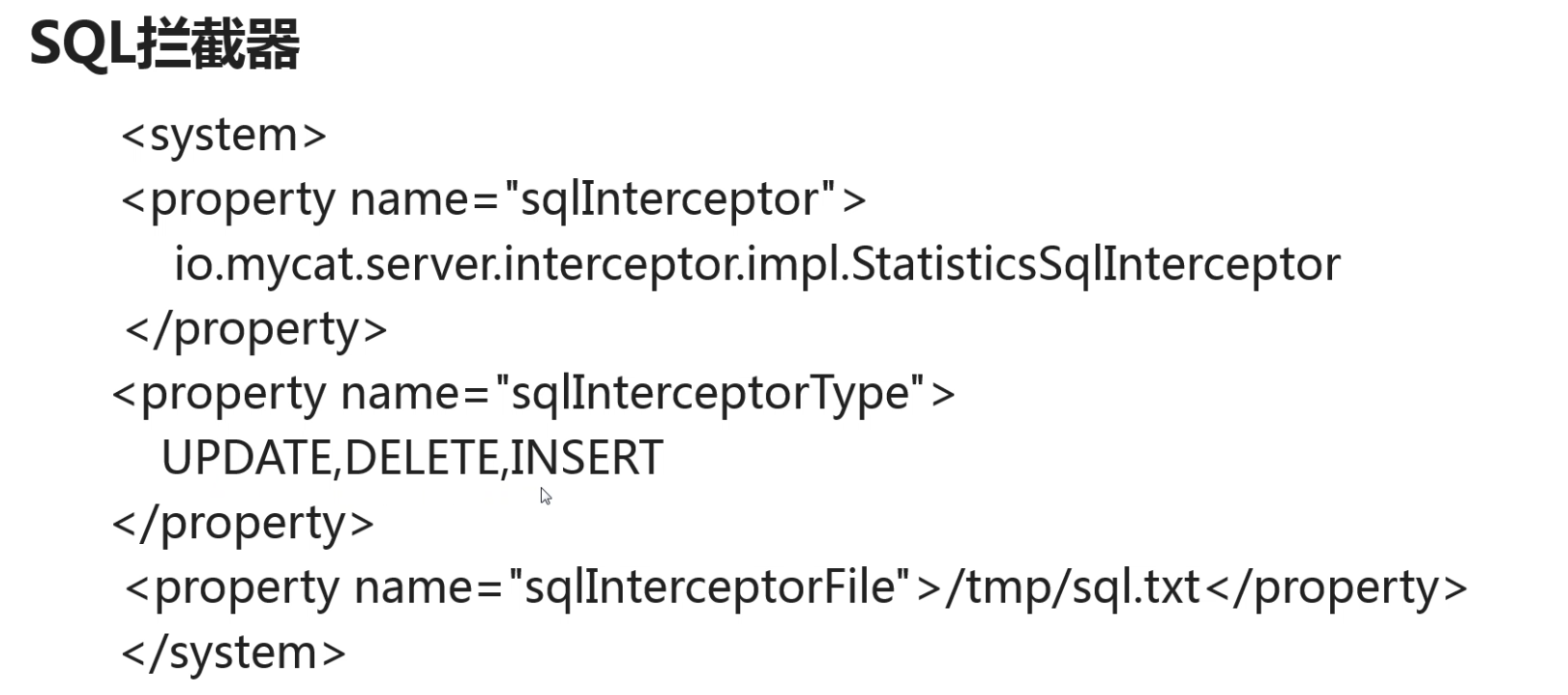
监控记录数据库写入操作

SQL审计

<property name="sqlInterceptor">io.mycat.server.interceptor.impl.StatisticsSqlInterceptor</property>

<property name="sqlInterceptorType">UPDATE,DELETE,INSERT</property>

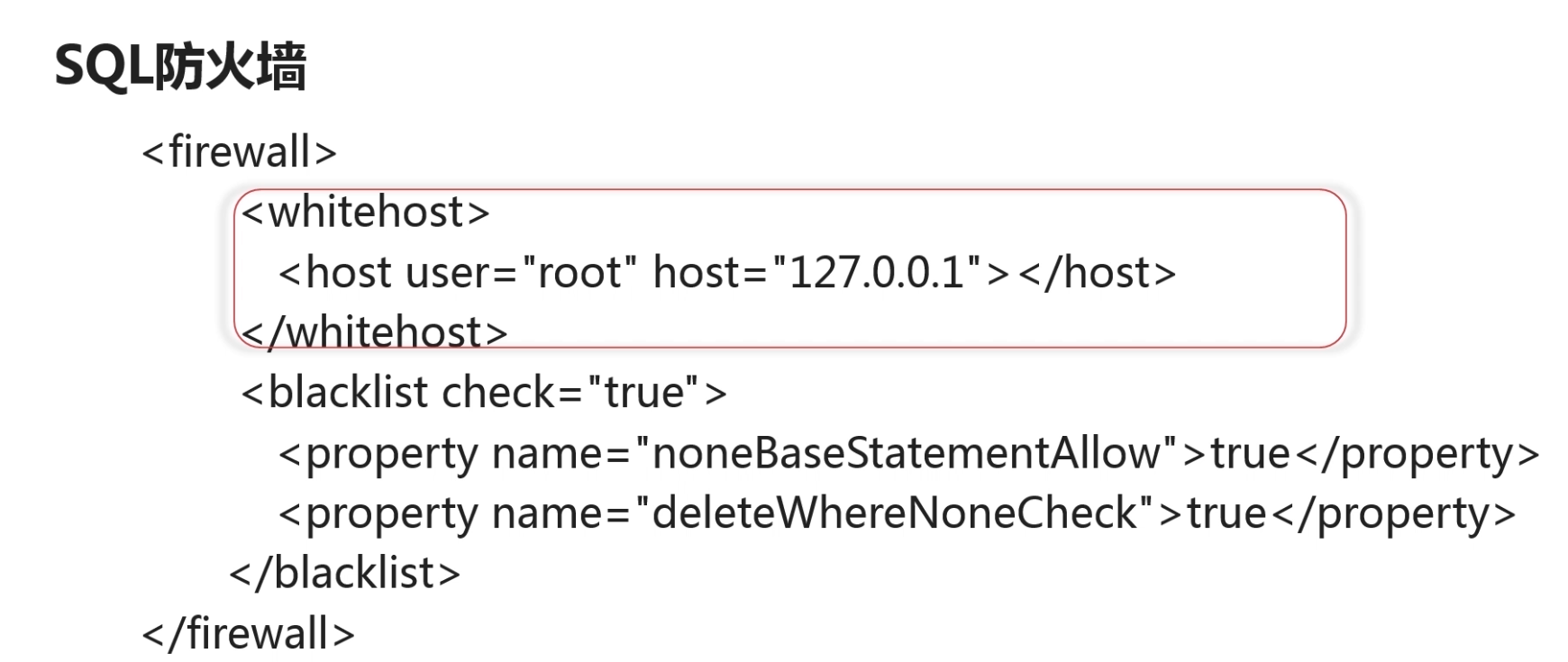
<property name="sqlInterceptorFile">/usr/local/mycat/sqltxt/sql.txt</property>



SQL防火墙

统一控制那些用户可以通过那些主机访问后端数据库

统一屏蔽一些sql语句。



## MyCAT高可用架构

开源映射地址

https://mirrors.shuosc.org/

高可用系统的基本要求

系统架构中不存在单点问题。

可以最大限度的保障服务的可用性。

使用zookeeper 配置mycat

1. 建立zookeeper集群
2. 初始化mycat配置到zk集群
3. 配置mycat支持zk启动
4. 启动mycat

|  |  |  |
| --- | --- | --- |
| 主机名 | Ip | 角色 |
| Node-1 | 10.10.2.45 2 | Mycat,zk,mysql |
| Node-2 | 10.10.2.40 3 | Zk,mysql |
| Node-3 | 10.10.2.43 4 | Zk,mysql |
| Node-4 | 10.10.2.\* 5 | Mycat,mysql |

wget 下载zookeeper

node1 节点下创建data 目录 创建 echo 0 >data/myid

node2 节点下创建data 目录 创建 echo 1 >data/myid

node3 节点下创建data 目录 创建 echo 2 >data/myid

编辑 conf/zoo.cfg 目录添加

dataDir=/sofeware/zookeeper/data

dataLogDir=/sofeware/zookeeper/logs

server.0=10.10.2.45:2888:3888

server.1=10.10.2.40:2888:3888

server.2=10.10.2.43:2888:3888

**初始化zk并配置mycat支持zk**

mycat bin 目录下 init\_zk\_data.sh 脚本

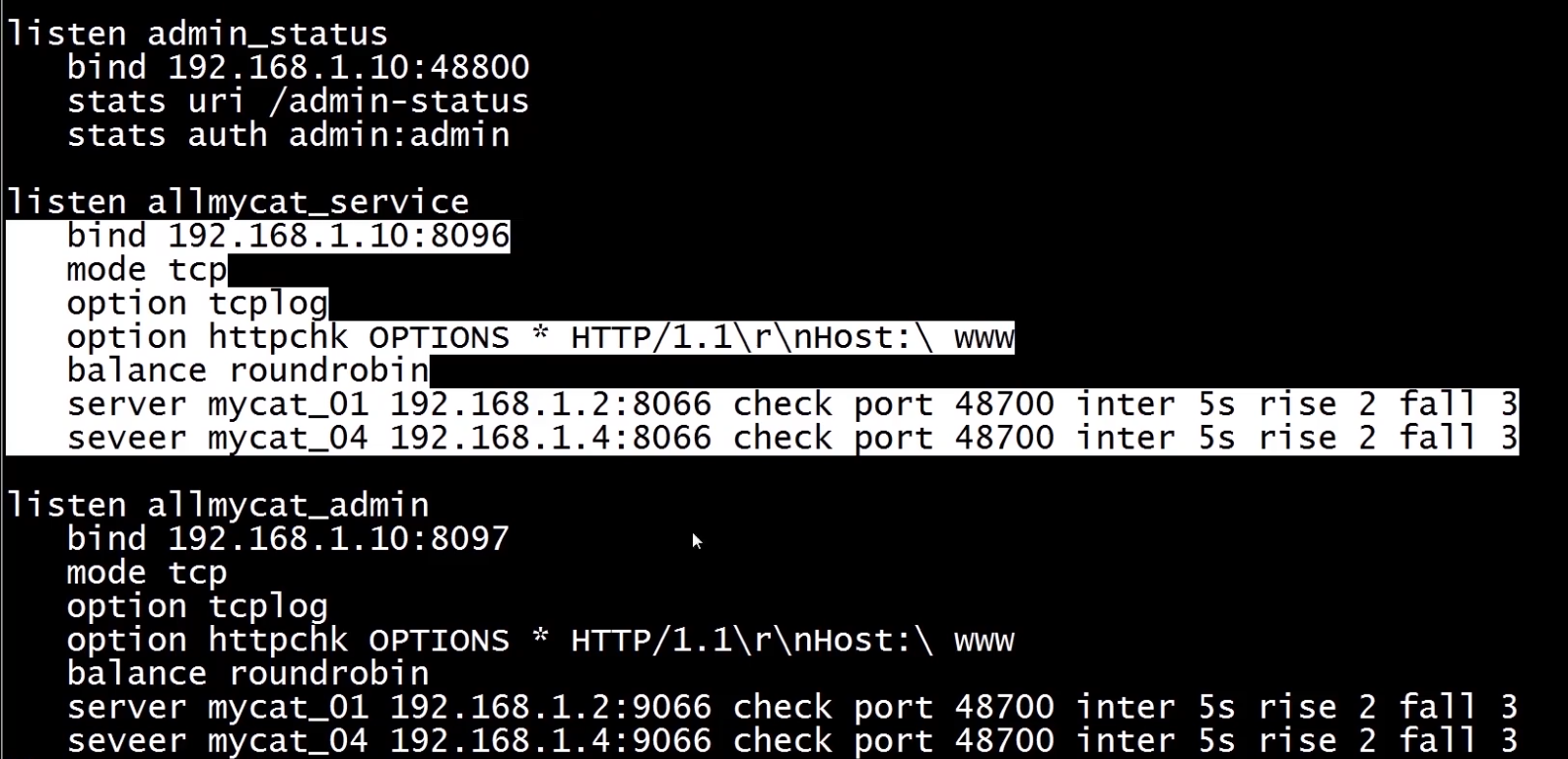
把mycat使用的配置文件复制\*.xml 到zkconf/ 下

cp schema.xml server.xml rule.xml sequence\_db\_conf.properties

## 使用Haproxy对Mycat负载均衡

1. 安装HAProxy

yum install haproxy –y



cd /etc/haproxy/

vim haproxy.cfg

listen admin\_status

bind 10.10.2.45:48800

stats uri /admin-status

stats auth admin:admin

#配置服务端口

listen allmycat\_service

bind 10.10.2.45:8096

mode tcp

option tcplog

option httpchk OPTIONS \* HTTP/1.1\r\nHost:\ www

balance roundrobin

server mycat\_01 10.10.2.45:8066 check port 48700 inter 5s rise 2 fall 3

server mycat\_38 10.10.2.38:8066 check port 48700 inter 5s rise 2 fall 3

#配置管理端口

listen allmycat\_admin

bind 10.10.2.45:8097

mode tcp

option tcplog

option httpchk OPTIONS \* HTTP/1.1\r\nHost:\ www

balance roundrobin

server mycat\_01 10.10.2.45:9066 check port 48700 inter 5s rise 2 fall 3

server mycat\_38 10.10.2.38:9066 check port 48700 inter 5s rise 2 fall 3

安装 yum install xinetd –y 守护进程服务 安装在Haproxy的服务器上。

1. 使用keepalived 监控Haproxy
2. 配置Haproxy监控Mycat
3. 配置应用通过vip访问Haproxy

|  |  |  |
| --- | --- | --- |
| 主机名 | Ip | 角色 |
| Node-1 | 10.10.2.45 2 | Mycat,zk,mysql,Haproxy,Keepalived |
| Node-2 | 10.10.2.40 3 | Zk,mysql |
| Node-3 | 10.10.2.43 4 | Zk,mysql |
| Node-4 | 10.10.2.38 | Mycat,mysql,Haproxy,Keepalived |